

Microservices/

Intro & SOA

micro-23

Aliaksei Bialiauski

Designed in L^AT_EX

All visual and text materials presented in this slidedeck are either originally made by the author or taken from public Internet sources, such as website. Copyright belongs to their respected authors.



Service-Oriented Architecture (SOA)

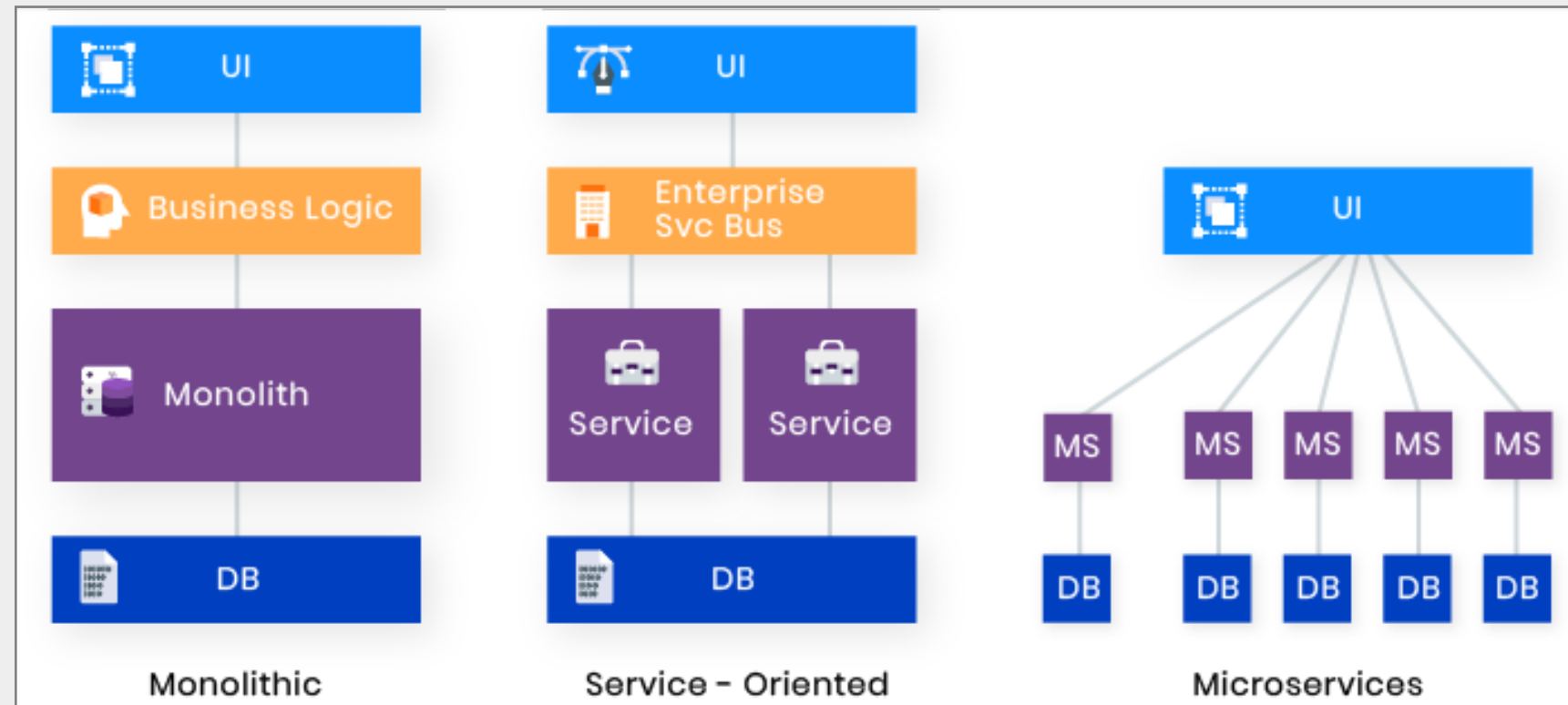
CAP theorem

Containers

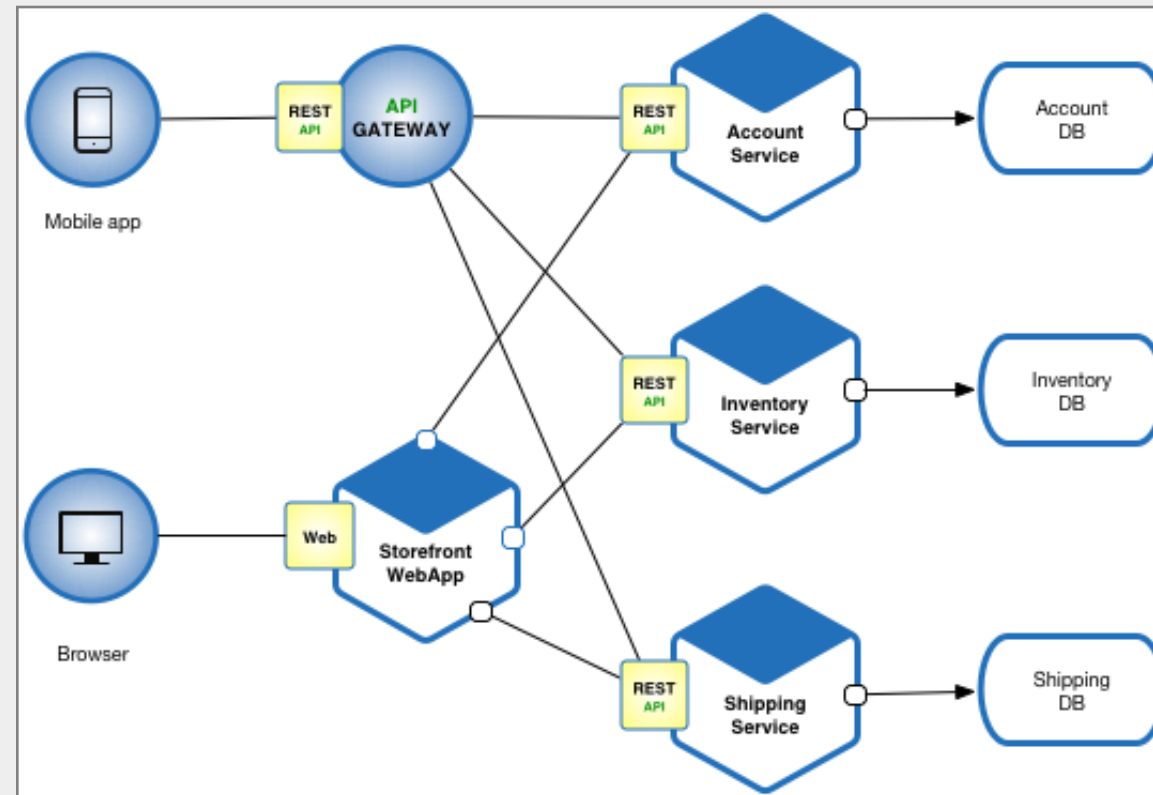
Chapter #1:

Service-Oriented Architecture (SOA)

Long time ago



Microservices



Microservices are a modern interpretation of service-oriented architectures used to build distributed software systems. — Wikipedia

API size

GitHub RESTful API

YouTube

Stateless vs. Stateful Architecture

“A stateless process or application can be understood in isolation. There is no stored knowledge of or reference to past transactions. Each transaction is made as if from scratch for the first time.” — RedHat

Chapter #2:

CAP theorem

Consistency

ACID vs BASE

Consistency model - rules that define the order of updates in the system and when these updates become visible to users. — Wikipedia

[Consistency Availability Network Partition]

Availability

$$A = \frac{E_{\text{up}}}{E_{\text{down}} + E_{\text{up}}}$$

Goal: minimize downtime

98% 99% 100%

it is not about a number

it is about architecture(static) and process(dynamic)

Partition tolerance

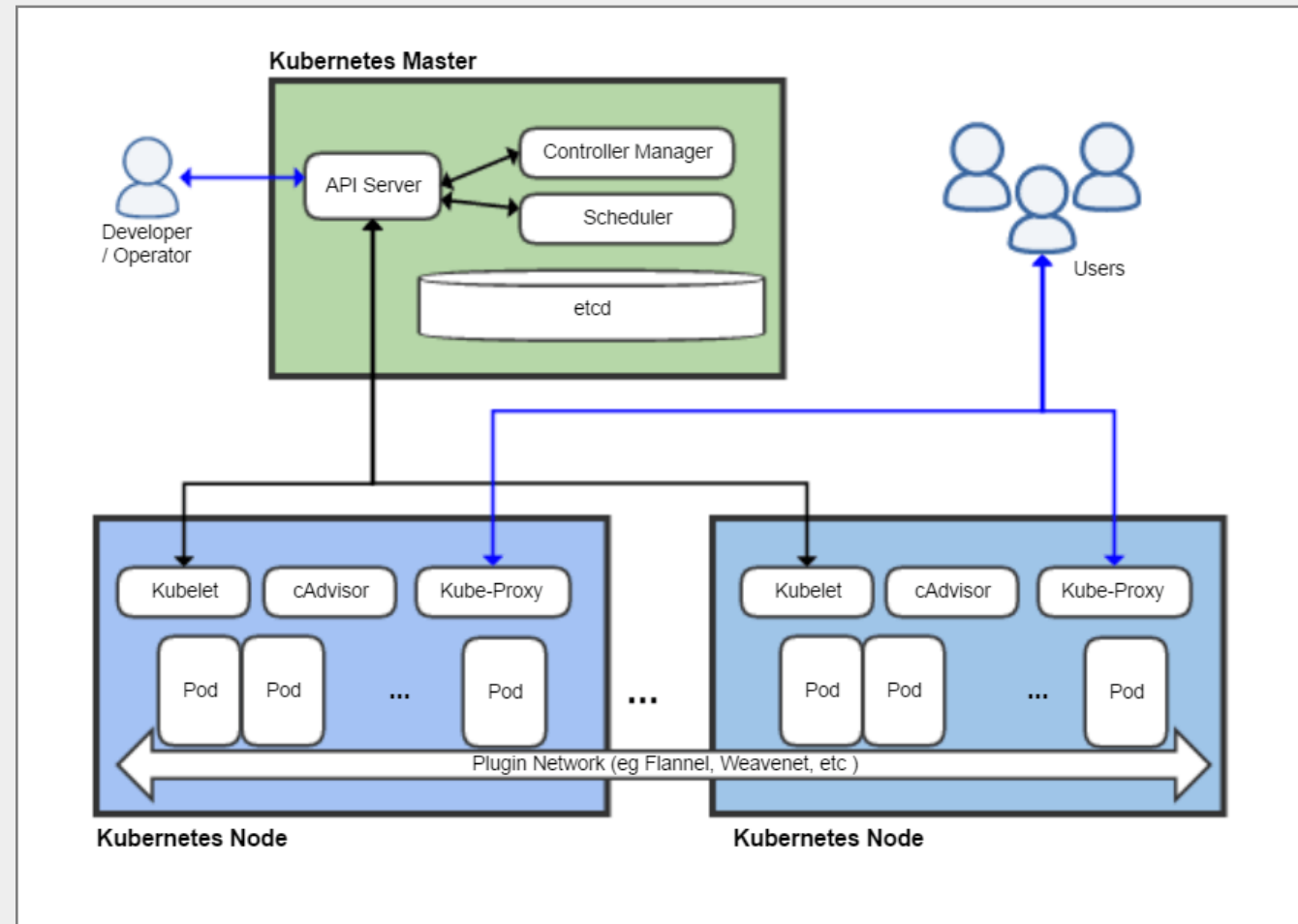
Partition tolerance, in the CAP context, means the ability of a data processing system to continue processing data even if a network partition causes communication errors between subsystems.

Chapter #3: Containers

Docker

```
1 FROM bellsoft/liberica-openjdk-alpine:11 as build
2 WORKDIR application
3 ARG JAR_FILE=target/*.jar
4 COPY ${JAR_FILE} app.jar
5 RUN java -Djarmode=layertools -jar app.jar extract
6 FROM bellsoft/liberica-openjdk-alpine:11
7 ENV TZ=Europe/Minsk
8 RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone
9 WORKDIR application
10 COPY --from=build application/dependencies/ ./
11 COPY --from=build application/spring-boot-loader/ ./
12 COPY --from=build application/snapshot-dependencies/ ./
13 RUN true
14 COPY --from=build application/application/ ./
15 ENTRYPOINT ["java", "org.springframework.boot.loader.JarLauncher"]
```

Kubernetes



Service Mesh

